We claim:

1. A method for virtualizing super-user privileges in a computer operating system including multiple virtual processes, the method comprising:

designating a plurality of virtual super-users, each virtual super-user being associated with a separate virtual process;

intercepting a system call for which actual super-user privileges are required;

in response to the intercepted system call being made by a virtual super-user and pertaining to the virtual process of the virtual super-user:

granting actual super-user privileges to the virtual super-user; and

allowing execution of the system call.

2. The method of claim 1, further comprising:

withdrawing the actual super-user privileges from the virtual super-user after execution of the system call.

3. The method of claim 1, wherein designating comprises:

assigning a virtual super-user identifier to each virtual super-user.

4. The method of claim 3, wherein each virtual super-user identifier comprises a super-user identifier and an indication of a virtual process.

5. The method of claim 1, wherein designating comprises:

assigning a user identifier to a virtual super-user; and

storing the user identifier and an indication of the virtual process of the virtual super-user in a virtual super-user list.

6. The method of claim 1, wherein granting comprises:

assigning a super-user identifier to the virtual super-user.

7. The method of claim 1, wherein the intercepted system call comprises a system call for accessing a file.

8. The method of claim 7, wherein the intercepted system call pertains to the virtual process of the virtual super-user when the file to be accessed is associated with the same virtual process.

9. The method of claim 1, wherein the intercepted system call comprises a system call for terminating a process.

10. The method of claim 9, wherein the intercepted system call pertains to the virtual process of the virtual super-user when the process to be terminated is associated with the same virtual process.

11. The method of claim 1, wherein the intercepted system call comprises a system call for terminating all processes associated with a virtual process, the method further comprising:

identifying each process associated with the virtual process; and

terminating each identified process.

12. The method of claim 11, wherein an association data structure stores associations between processes and virtual processes, and wherein identifying comprises:

identifying each process by its association with the virtual process in the association data structure.

13. The method of claim 1, wherein the system call is made by a virtual super-user when a user making the call has a virtual super-user identifier.

14. The method of claim 1, wherein the system call is made by a virtual super-user when a user making the call has user identifier in a virtual super-user list.

15. The method of claim 1, further comprising:

responsive to the intercepted system call not being made by a virtual super-user, disallowing execution of the system call.

16. The method of claim 1, further comprising:

responsive to the intercepted system call being made by a virtual super-user and not pertaining to the virtual process of the virtual super-user, disallowing execution of the system call.

17. The method of claim 1, further comprising:

responsive to the intercepted system call comprising a system call for inserting a module into an operating system kernel, disallowing execution of the system call.

18. The method of claim 1, wherein allowing comprises:

executing the system call.

19. The method of claim 1, wherein intercepting a system call comprises:

loading a system call wrapper;

saving a pointer to the system call; and

replacing the pointer to the system call with a pointer to the system call wrapper, such that the system call wrapper is executed when the system call is invoked.

20. The method of claim 19, wherein the pointer to the first system call comprises a system call vector.

21. A computer program product for virtualizing super-user privileges in a computer operating system including multiple virtual processes, the computer program product comprising:

program code for designating a plurality of virtual super-users, each virtual super-user being associated with a separate virtual process;

program code for intercepting a system call for which actual super-user privileges are required;

program code for determining that the intercepted system call was made by a virtual super-user and pertains to the virtual process of the virtual super-user; granting actual super-user privileges to the virtual super-user; and allowing execution of the system call.

22. The computer program product of claim 21, further comprising:

program code for withdrawing the actual super-user privileges from the virtual super-user after execution of the system call.

23. The computer program product of claim 21, wherein program code for designating comprises:

program code for assigning a virtual super-user identifier to each virtual super-user.

24. The computer program product of claim 23, wherein each virtual super-user identifier comprises a super-user identifier and an indication of a virtual process.

25. The computer program product of claim 21, wherein program code for designating comprises:

program code for assigning a user identifier to a virtual super-user; and

program code for storing the user identifier and an indication of the virtual process of the virtual super-user in a virtual super-user list.

26. The computer program product of claim 21, wherein program code for granting comprises:

program code for assigning a super-user identifier to the virtual super-user.

27. The computer program product of claim 21, wherein the intercepted system call comprises a system call for accessing a file.

28. The computer program product of claim 27, wherein the intercepted system call pertains to the virtual process of the virtual super-user when the file to be accessed is associated with the same virtual process.

29. The computer program product of claim 21, wherein the intercepted system call comprises a system call for terminating a process.

30. The computer program product of claim 29, wherein the intercepted system call pertains to the virtual process of the virtual super-user when the process to be terminated is associated with the same virtual process.

31. The computer program product of claim 21, wherein the intercepted system call comprises a system call for terminating all processes associated with a virtual process, the computer program product further comprising:

program code for identifying each process associated with the virtual process; and

program code for terminating each identified process.

32. The computer program product of claim 31, wherein an association data structure stores associations between processes and virtual processes, and wherein program code for identifying comprises:

program code for identifying each process by its association with the virtual process in the association data structure.

33. The computer program product of claim 21, wherein the system call is made by a virtual super-user when a user making the call has a virtual super-user identifier.

34. The computer program product of claim 21, wherein the system call is made by a virtual super-user when a user making the call has a user identifier in a virtual super-user list.

35. The computer program product of claim 21, further comprising:

program code for disallowing execution of the system call in response to the intercepted system call not being made by a virtual super-user.

36. The computer program product of claim 21, further comprising:

program code for disallowing execution of the system call in response to the intercepted system call being made by a virtual super-user and not pertaining to the virtual process of the virtual super-user.

37. The computer program product of claim 21, further comprising:

program code for disallowing execution of the system call in response to the intercepted system call comprising a system call for inserting a module into an operating system kernel.

38. The computer program product of claim 21, wherein program code for allowing comprises:

program code for executing the system call.

39. The computer program product of claim 21, wherein program code intercepting a system call comprises:

program code for loading a system call wrapper;

program code for saving a pointer to the system call; and

program code for replacing the pointer to the system call with a pointer to the system call wrapper, such that the system call wrapper is executed when the system call is invoked.

40. The computer program product of claim 19, wherein the pointer to the first system call comprises a system call vector.

41. A system for virtualizing super-user privileges in a computer operating system including multiple virtual processes, the system comprising:

a virtual super-user designation module for designating a plurality of virtual super-users, each virtual super-user being associated with a separate virtual process; and

a system call wrapper for intercepting a system call for which actual super-user privileges are required and, in response to the intercepted system call being made by a virtual super-user and pertaining to the virtual process of the virtual super-user, granting actual super-user privileges to the virtual super-user and allowing execution of the system call.

42. The system of claim 41, wherein the system call wrapper is further configured to withdraw the actual super-user privileges from the virtual super-user after execution of the system call.

43. The system of claim 41, wherein the virtual super-user designation module is further configured to assign a virtual super-user identifier to each virtual super-user.

44. The system of claim 43, wherein each virtual super-user identifier comprises a super-user identifier and an indication of a virtual process.

45. The system of claim 41, wherein the virtual super-user designation module is further configured to assign a user identifier to a virtual super-user and store the user identifier and an indication of the virtual process of the virtual super-user in a virtual super-user list.

46. The system of claim 41, wherein the system call wrapper is further configured to assign a super-user identifier to the virtual super-user.

47. The system of claim 41, wherein the intercepted system call comprises a system call for accessing a file.

48. The system of claim 47, wherein the intercepted system call pertains to the virtual process of the virtual super-user when the file to be accessed is associated with the same virtual process.

49. The system of claim 41, wherein the intercepted system call comprises a system call for terminating a process.

50. The system of claim 49, wherein the intercepted system call pertains to the virtual process of the virtual super-user when the process to be terminated is associated with the same virtual process.

51. The system of claim 41, wherein the intercepted system call comprises a system call for terminating all processes associated with a virtual process, and wherein the system call wrapper is further configured to identify each process associated with the virtual process and terminate each identified process.

52. The system of claim 51, further comprising:

an association data structure for storing associations between processes and virtual processes, wherein the system call wrapper is further configured to identify each process by its association with the virtual process in the association data structure.

53. The system of claim 41, wherein the system call is made by a virtual super-user when a user making the call has a virtual super-user identifier.

54. The system of claim 41, wherein the system call is made by a virtual super-user when a user making the call has user identifier in a virtual super-user list.

55. The system of claim 41, wherein the system call wrapper is further configured to disallow execution of the intercepted system call in response to the intercepted system call not being made by a virtual super-user.

56. The system of claim 41, wherein the system call wrapper is further configured to disallow execution of the intercepted system call in response to the intercepted system call being made by a virtual super-user and not pertaining to the virtual process of the virtual super-user.

57. The system of claim 41, wherein the system call wrapper is further configured to disallow execution of the intercepted system call in response to the intercepted system call comprising a system call for inserting a module into an operating system kernel.

58. The system of claim 41, wherein the system call wrapper is further configured to execute the system call.